Community Evolution Detection in Dynamic Heterogeneous Information Networks *

Yizhou Sun
UIUC
Urbana, IL, USA
sun22@uiuc.edu

Jie Tang
Tsinghua University
Beijing, China
jietang@tsinghua.edu.cn

Jiawei Han
UIUC
Urbana, IL, USA
hanj@cs.uiuc.edu

Manish Gupta
UIUC
Urbana, IL, USA
gupta58@uiuc.edu

Bo Zhao
UIUC
Urbana, IL, USA
bozhao3@uiuc.edu

ABSTRACT

As the rapid development of all kinds of online databases, huge heterogeneous information networks thus derived are ubiquitous. Detecting evolutionary communities in these networks can help people better understand the structural evolution of the networks. However, most of the current community evolution analysis is based on the homogeneous networks, while a real community usually involves different types of objects in a heterogeneous network. For example, when referring to a research community, it contains a set of authors, a set of conferences or journals and a set of terms.

In this paper, we study the problem of detecting evolutionary *multi-typed* communities defined as net-clusters in dynamic heterogeneous networks. A Dirichlet Process Mixture Model-based generative model is proposed to model the community generations. At each time stamp, a clustering of communities with the best cluster number that can best explain the current and historical networks are automatically detected. A Gibbs sampling-based inference algorithm is provided to inference the model. Also, the evolution structure can be read from the model, which can help users better understand the birth, split and death of communities. Experiments on two real datasets, namely DBLP and Delicious.com, have shown the effectiveness of the algorithm.

1. INTRODUCTION

As the rapid development of all kinds of online databases, huge heterogeneous information networks thus derived are ubiquitous, which contain different types of objects. Examples include online bibliographic databases such as DBLP[1], social websites such as Flickr[2], and tagging websites such as Delicious[3]. The networks derived contain different types of objects, which are different from traditional homogeneous networks, such as friendship networks and co-author networks. Detecting evolutionary communities from these heterogeneous networks will benefit the users of these online databases better understanding the structures of the complex networks and their evolution along with time. Also, such knowledge will help users make good predictions on the future trends of the community. In contrast to community defined in a homogeneous network, which is a set of objects from a single type, a community in a heterogeneous network should be heterogeneous itself. Using the bibliographic network as an example, the communities we are interested in are research areas, which contain objects of authors, venues and terms. We call such multi-typed communities *net-clusters*, following our previous work [19].

However, most existing methods only study the community evolution in *homogeneous networks*. The traditional network evolution analysis on homogeneous networks, which is only able to track one type of objects' evolution, cannot correctly model the real evolution of a community that actually contains multiple types of objects. For example, if we only study the evolution of co-author network extracted from the DBLP bibliographic network, we may make two mistakes: (1) detect false research communities by considering the newly joined authors with existing research interests as new communities; or (2) miss new research communities by considering the same group of authors with new research topics as an old community. Other recent works on evolution study on *heterogeneous networks*, such as in [20], have considered the interaction of communities among different types, however, their community definition is still single-typed, and cannot reflect the concept of multi-typed communities like research areas.

By intuition, a good evolutionary model of communities

*Research was sponsored in part by the U.S. National Science Foundation under grant IIS-09-05215, and by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

MLG '10, July 24-25, 2010 Washington, DC, USA

[1] http://www.informatik.uni-trier.de/∼ley/db/

[2] http://www.flickr.com/

[3] http://delicious.com/

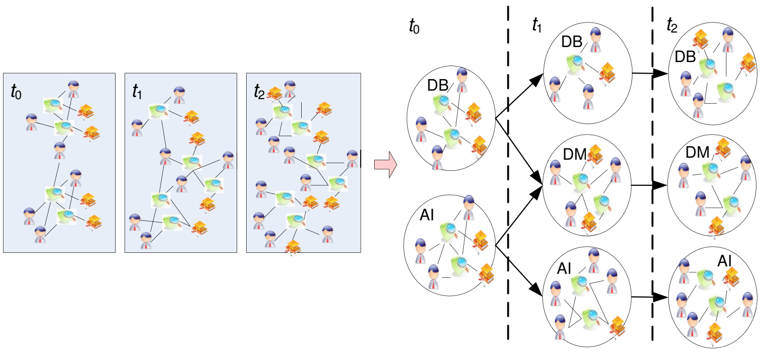


Figure 1: Illustration of Detecting Evolutionary Net-Clusters

should contain two properties: (1) the number of communities in each time stamp should be flexible and automatically learned; (2) the communities in adjacent timestamps should be consistent. In this paper, we propose a novel method to model the community evolution in heterogeneous information networks. First, we use net-clusters which are designed for heterogeneous networks to model the communities. Each net-cluster contains different types of objects and follows the same schema of the original networks. Second, a Dirichlet Process Mixture (DPM) Model-based net-cluster generative model (Evo-NetClus) is proposed to model the generation process of net-clusters at each time window, which is able to decide the natural cluster number and consider historical impacts from net-clusters of previous time windows simultaneously. By solving this model, net-clusters that with the best cluster number and best consistent with the historical net-clusters are generated. Explicit evolution structure can thus be obtained from the prior dependency among different net-clusters between adjacent timestamps. The problem of evolutionary community detection process is illustrated in Figure 1. The experiments using the bibliographic network extracted from DBLP and Delicious.com have shown the effectiveness of our method.

The major contributions of this paper are:

1. Propose the problem to detect multi-typed evolutionary communities in a heterogeneous network.

2. Propose a novel DPM model-based generative model (Evo-NetClus) to model the generation of net-clusters, which can automatically learn the best cluster number, and keep consistency between adjacent net-clusters. A Gibbs sampling-based method is proposed to inference the model.

3. We apply the method on two real datasets, the DBLP network and Delicious network, and the results show the power of our model which are able to use both heterogeneous information and time information of the networks.

2. RELATED WORK

Community detection and clustering in networks have been studied for quite a long time, which is trying to split a gigantic network into several relatively independent parts and group similar nodes into the same clusters. The study of community detection problem is first on homogeneous networks, such as spectral clustering methods [16, 22, 23], modularity-based methods [13, 12], and probabilistic model-based methods [17, 8, 7, 1], and later to bipartite networks [26, 5], and recently on heterogeneous networks [18, 19]. In this paper, we will consider the heterogeneous networks with star network schema as in [19], which is a very popular case in real world. However, different from our previous work and other static community detection methods, this study mainly focuses on how to model the dynamic evolution of the net-cluster-based multi-typed communities.

In practice, new nodes will join in the network, while some nodes will leave, and thus a sequence of networks with different timestamps can be collected from dynamic evolving networks. Detecting evolutionary clusters on such network sequences can help people better understand the evolution of communities. Some studies have been devoted on homogeneous networks, extended from static clustering methods, such as in [4, 20, 9, 15, 6]. A recent work [20] studied the evolution on heterogeneous networks. However, their communities are defined on each single type of objects, and cluster numbers for each type need to be fixed and specified by users. While our community evolution studies a more comprehensive concept of community, which can also automatically decide the cluster numbers.

The most challenging problem in studying community evolution is how to decide the correct cluster number in each timestamp. In this paper, we propose a Dirichlet Process Mixture Model-based generative model to detect evolutionary net-clusters in heterogeneous networks with star network schema, which is able to model the cluster number in each timestamp and smoothness between net-clusters from consecutive timestamps simultaneously. Dirichlet Process [11, 21] provides a way to add priors to the cluster number for mixture models, and thus is very helpful to decide the cluster number automatically. Recently, some works have extended the Dirichlet Process into considering time information, such as in [27] and [14]. Some other DP-based extensions [24, 25] have been proposed to model evolutionary clustering. The differences of our model from these methods include: (1) we provide a specific solution to net-cluster evolution in heterogeneous networks; (2) we defined a novel generative model for net-cluster evolution, which can model the evolution of the same cluster in different timestamps, while many existing works require the same clusters (atom distributions) do not change among different timestamps; (3) we do not claim a global inference of the model, but greedy inference at each time stamp, which is more practical for timely updating the evolution.

Another related work to evolutionary network clustering is evolutionary topic modeling, which tries to extract the best topic models in each timestamp that satisfy constraints of temporal smooth, such as in [3, 10]. However, merely studying the evolution of topics without considering the link information in the networks cannot fully reflect the evolution of communities. On one hand, a community containing different types of objects is more meaningful and useful; on the other hand, links in the network can tell more about the connection between clusters and can help detect more accurate evolutionary clusters. In this paper, terms are treated as objects of one of the attribute types and evolutionary net-clusters are instead modeled and detected.

3. PRELIMINARIES

In this section, we will introduce some definitions and preliminary knowledge of this work, including the concept of net-cluster and the Dirichlet Process Mixture (DPM) Model.

3.1 Net-Cluster

Definition 1. **Information Network.** An *information network* is defined as $G = (V, E, W; \mathcal{A}, \psi)$, where V is the vertex set, $E \subseteq V \times V$ is the link set, $W : E \rightarrow R^+$ is a weight function defined on E, $\mathcal{A}$ is an alphabetical table denoting names of different types of vertices, and $\psi : V \rightarrow \mathcal{A}$ is the mapping from each vertex to its type. If the number of types $|A| > 1$, G is called a **heterogeneous information network**; otherwise, G is a **homogeneous information network**.

A net-cluster is the cluster defined on information networks.

Definition 2. **Net-cluster.** Given a network $G = \langle V, E, W\rangle$, a net-cluster C is defined as $C = \langle G', p_C\rangle$, where G' is a **sub-network** of G, *i.e.*, $V(G') \subseteq V(G)$, $E(G') \subseteq E(G)$, and $\forall e_{ij} = \langle x_i, x_j\rangle \in E(G'), W(G')_{x_i x_j} = W(G)_{x_i x_j}$. Function $p_C : V(G') \rightarrow [0, 1]$ is defined on $V(G')$, for all $x \in V(G')$, $0 \leq p_C(x) \leq 1$, which denotes the probability that x belongs to cluster C, *i.e.*, $P(x \in C)$.

From topology point of view, a net-cluster is a sub-network of the original network, following the same network schema. In this paper, we only consider the networks with star network schema, *i.e.*, links only exist between center type of objects (*e.g.*, papers), which we called target objects, and several other types of objects (*e.g.*, authors, conferences, and terms), which we called attribute objects. At the same time, a net-cluster is attached with statistical information that describes the net-cluster. The statistical information includes the posterior probability that each object belongs to the net-cluster, which is defined as p_C in Definition 2, and ranking distribution $P_{\psi(x)}(X|G_k)$ ([18, 19]) for attribute objects x from the type $\psi(x)$, which denotes the probability of object $x \in X$ appearing in the net-cluster G_k. Finally, each net-cluster G_k is also referring to a statistical model that defines the probability $p(o_i|G_k)$ to generate a target object o_i, given the attribute objects linked to it, in the net-cluster G_k.

To better illustrate the idea of the evolutionary net-cluster problem and the algorithms, we will use the bibliographic network extracted from DBLP as an example. However, the algorithm can be used in any heterogeneous network with a star network schema. In the bibliographic network derived from DBLP, there are four types of objects, namely the papers (O), the authors (A), the venues (conferences and journals) (C), and the terms (D), where links are only existing between papers and the remaining three types of objects. For each target object (paper) $o_i \in O$, it can be represented as a tuple $(\mathbf{a}_i, \mathbf{c}_i, \mathbf{d}_i)$, where $\mathbf{a}_i = (a_{i1}, \ldots, a_{i|A|})$ is a vector with length of $|A|$ and a_{ij} denotes the weight of the link between o_i and author $a_j \in A$. $\mathbf{c}_i$ and $\mathbf{d}_i$ are with similar meanings, which are the conference vector and term vector associated with paper o_i.

Let ϕ_k be the statistical model associating with net-cluster G_k (thereafter k for short), and $\theta_k = (\theta_k^A, \theta_k^C, \theta_k^D)$ be the parameters of the cluster model, where θ_k^A represents the conditional ranking distributions of objects from Type A in net-cluster k, *i.e.*, $\theta_k^A(a) = p_A(A = a|k)$ (similarly for θ_k^C and θ_k^D). The probability for generating o_i in net-cluster k is then $p(o_i|k) = p(\mathbf{a}_i|k)p(\mathbf{c}_i|k)p(\mathbf{d}_i|k)$, where $p(\mathbf{a}_i|k) = \prod_{j=1}^{|A|} p_A(a_j|k)^{a_{ij}}$, which is a multinomial distribution, and similarly for $p(\mathbf{c}_i|k)$ and $p(\mathbf{d}_i|k)$.

Definition 3. **Network Sequence.** A dynamic network sequence $\mathcal{GS} = (G_1, G_2, \ldots, G_t, \ldots)$ is a sequence of networks, where $G_t = \langle V_t, E_t, W_t\rangle$ is the network associated with timestamp t.

Objects in the networks are usually associated with time. For example, papers are associated with publication years in bibliographic networks. A dynamic network can then be extracted into network sequence according to such time information. In the DBLP case, each G_t is a network comprised of all the papers published in year t, as well as all the authors, conferences and terms linking to them. In the Delicious Case, each G_t is a network comprised of all the tagging events happened in the time window t, and all the users, websites and tags associated with these tagging events.

3.2 Dirichlet Process Mixture Model

Mixture model is a frequently used method in clustering, which assumes an observation o_i is generated from a fixed number, say K, of different statistical models $\{\phi_k\}_{k=1}^K$ (clusters), with different component weights π_k. By maximizing the log-likelihood of all the observations, both the component weights and the parameters for each cluster are obtained, and a soft clustering can be achieved accordingly. A mixture model can be formalized as

$$o_i \sim \sum_{k=1}^{K} \pi_k p(o_i|z_i = k) \tag{1}$$

where z_i denotes the hidden cluster label associated with object o_i.

However, it is usually difficult for people to specify the correct cluster number K in the mixture model. Dirichlet Process Mixture Model is a typical way to solve the problem, where the cluster number is considered as countable infinite, and the distribution of component weights follows a Dirichlet Process (an extension of Dirichlet Distribution to infinite space) with a base distribution G_0. We follow the work [11] and define the DPM model as in Eq. (2):

$$\begin{aligned} o_i|\theta_i &\sim f(\theta_i) \\ \theta_i|G &\sim G \\ G &\sim \mathcal{DP}(G_0, \alpha) \end{aligned} \tag{2}$$

where θ_i is the parameter of the cluster associated with o_i and it follows the distribution of G. The distribution G is generated by a Dirichlet Process with the base measure αG_0. α is the concentration parameter.

According to [11], this model is equivalent to the following finite mixture models (Eq. (3)), with the cluster number K goes to infinity:

$$\begin{aligned} o_i | z_i, \{\theta_k\}_{k=1}^{K} &\sim f(\theta_{z_i}) \\ z_i | \pi &\sim Discrete(\pi_1, \ldots, \pi_K) \\ \theta_k &\sim G_0 \\ \pi &\sim Dirichlet(\alpha/K, \ldots, \alpha/K) \end{aligned} \tag{3}$$

where z_i stands for the latent class label of the observation o_i. In this model, given the cluster number K, the parameters for all the clusters are drawn from the same prior distribution G_0, and the component weights are drawn from a Dirichlet Distribution as the prior. In Section 4, we will show why and how the DPM model can be extended and used to model the generation and evolution of net-clusters, where different net-clusters may be generated using different priors and the evolutionary structure can be build explicitly accordingly.

4. EVOLUTIONARY NET-CLUSTER DETECTION

In this section, we will introduce the method of detecting evolutionary net-clusters in a heterogeneous network using the DBLP bibliographic network as an example. In the DBLP data, the publication year of the papers naturally divides the original dynamic networks into different network shots. Each network contains all the papers published in one year, as well as the authors, venues and terms linked to these papers. Given the network sequence, without the need to specify the cluster number for each timestamp, our algorithm is able to output the best net-clusters that are not only consistent with the current timestamp network, but also consistent with the networks from the previous timestamps.

4.1 Evolutionary Net-Cluster Generative Model: Evo-NetClus

We now first introduce our novel DMP model-based generative model, Evo-NetClus, for evolutionary net-clusters. When considering the evolution of communities, different situations should be modeled: some minor changes may happen to an existing community, some new communities can be generated, some small communities may disappear, some communities will be split into several sub areas, and several communities may be merged into one big community. Therefore, it requires: (1) the number of clusters in each timestamp cannot be fixed and is impossible to be specified by users; (2) the generative model should be flexible to describe different statistical information under different timestamp for the same net-cluster; and (3) the evolution of net-clusters should be smooth between adjacent timestamps, and the smoothness should be modeled in the generative model as well. We therefore propose Evo-NetClus, whose graphical model is given in Figure 2. At each timestamp t, we build a DPM for the target objects $\{o_{i,t}\}_{i=1}^{N_t}$ in network G_t. For $t = 1$, namely the first timestamp, the target objects are modeled as a standard DPM:

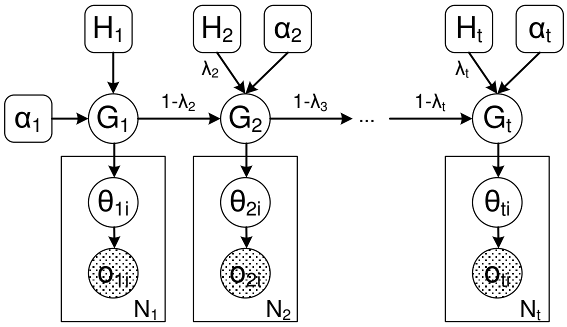


Figure 2: Graphical Model for Evo-NetClus

$$\begin{aligned} o_{i,t} | \theta_{i,t} &\sim f(\theta_{i,t}) \\ \theta_i | G_t &\sim G_t \\ G_t &\sim \mathcal{DP}(Dir(\beta_t H_t), \alpha_t) \end{aligned} \tag{4}$$

where $Dir(\beta_t H_t)$ is the base measure, which is a Dirichlet distribution with the expectation distribution as H_t and the precision β_t, and α_t is concentration parameter for G_t. For later networks G_t that $t > 1$, the target objects are also modeled as a DPM, but with the base measure for G_t a mixture model of Dirichlet distribution with the expectation distributions as the historical model G_{t-1} and a background model H_t respectively:

$$\begin{aligned} o_{i,t} | \theta_{i,t} &\sim f(\theta_{i,t}) \\ \theta_i | G_t &\sim G_t \\ G_t &\sim \mathcal{DP}(\lambda_t Dir(\beta_t H_t) + \sum_k (1 - \lambda_t) \pi_{k,t-1} Dir(\beta_t G_{k,t-1}), \alpha_t) \end{aligned} \tag{5}$$

where λ_t is the mixture portion of background.

We now specify the distributions used in Eq. 4 and 5 in their mixture model view as in Eq. 3. Suppose the proper but unknown number of net-clusters is K, let $\mathbf{\Theta}_t = \{\theta_{k,t}\}_{k=1}^{K}$, with each $\theta_{k,t} = (\theta_{k,t}^A, \theta_{k,t}^C, \theta_{k,t}^D)$. $\theta_{k,t}^A, \theta_{k,t}^C$ and $\theta_{k,t}^D$ denote the parameters for types A, C, and D respectively, where $\theta_{k,t}^A(j) = p_A(a_j | k, t)$ stands for the probability of object a_j in net-cluster $G_{k,t}$, $a_{ij,t}$ is the weight of link between o_i and a_j at timestamp t. Similar definitions are for $\theta_{k,t}^C(j)$, $\theta_{k,t}^D(j)$, $c_{ij,t}$ and $d_{ij,t}$. For each net-cluster $G_{k,t}$, the statistical model for target objects $o_{i,t} = (\mathbf{a}_{i,t}, \mathbf{c}_{i,t}, \mathbf{d}_{i,t})$ mentioned in Section 3.1 and originally from [19] is then:

$$\begin{aligned} p(o_{i,t} | z_{i,t} = k, \mathbf{\Theta}_t) &= p(o_{i,t} | \theta_{k,t}) \\ &= p(\mathbf{a}_{i,t} | \theta_{k,t}^A) p(\mathbf{c}_{i,t} | \theta_{k,t}^C) p(\mathbf{d}_{i,t} | \theta_{k,t}^D) \\ &= \prod_{j=1}^{|A|} \theta_{k,t}^A(j)^{a_{ij,t}} \prod_{j=1}^{|C|} \theta_{k,t}^C(j)^{c_{ij,t}} \prod_{j=1}^{|D|} \theta_{k,t}^D(j)^{d_{ij,t}} \end{aligned} \tag{6}$$

For the base measures $G_{0,t} = Dir(\beta_t H_t)$, they are independent symmetric Dirichlet distributions for parameters associated with each type, namely, $H_t(\mathbf{\Theta}_t) = (P(\theta_{k,t}^A), P(\theta_{k,t}^C), P(\theta_{k,t}^D))$, with each component defined as a uniform distribution over all the objects. Each component of the base measure $G_{0,t}(\mathbf{\Theta}_t)$

is then defined as:

$$\begin{aligned}\theta^A_{k,t} &\sim Dirichlet(\beta_A/|A|_t, \ldots, \beta_A/|A|_t)\\ \theta^C_{k,t} &\sim Dirichlet(\beta_C/|C|_t, \ldots, \beta_C/|C|_t)\\ \theta^D_{k,t} &\sim Dirichlet(\beta_D/|D|_t, \ldots, \beta_D/|D|_t)\end{aligned} \tag{7}$$

where β_A, β_B and β_C are precise for each Dirichlet distribution component, and $|A|_t$, $|C|_t$ and $|D|_t$ denote the cardinality of the objects in each type at timestamp t. Notice that, other reasonable distributions may also be used as base measures for $G_{0,t}$. We use symmetric Dirichlet distributions, because it can be served as conjugate priors for the conditional rankings of objects in each type, which follow multinomial distributions.

Instead of finding a global optimization on $\{\mathbf{\Theta}_t\}_{t=1}^T$, given a network sequence $\mathcal{GS} = (G_1, G_2, \ldots, G_T)$, as did in HMM inference and some dynamic Dirichlet Process model [14, 27], our goal is to find greedy optimization configuration $\mathbf{\Theta}_t$ based only on the history. Formally, we want to maximize the posterior probability $p(\mathbf{\Theta}_t|\mathbf{O}_t, \mathbf{\Theta}^*_{t-1})$, where $\mathbf{\Theta}^*_{t-1}$ is the best configuration for $\mathbf{\Theta}_{t-1}$ in timestamp $t-1$. We make such choice because usually we are not able to observe the ***whole*** network sequence. In practice, networks are coming in a stream way, with new objects emerging, which makes the global vocabulary unavailable. Also, we do not want to compute the ***whole*** network sequence when a new network comes, but be more willing to use new data to update the model sequentially. Another reason is the global inference could be rather complex and intractable.

We now show the generative model Evo-NetClus satisfies the three requirements proposed in the beginning of this section, by using the DPM model and carefully designing the prior distribution for each timestamp t as a mixture over the historical model in $t-1$ and a new background model.

4.1.1 Modeling Flexible Cluster Number

First, a DP defines a prior probability over the structure of the clusters. Let $z_1, z_2, \ldots, z_{i-1}$ are hidden cluster labels for existing objects $o_1, o_2, \ldots, o_{i-1}$, according to DP, the conditional probabilities for a new object o_i to join a existing net-cluster or to create a new net-cluster k are:

- if k is an existing net-cluster, *i.e.*, $z_i = z_j$ for some $j < i$, the probability to generate o_i from net-cluster k is then $\frac{n_k}{\alpha+i-1}$;
- if k is a new net-cluster, *i.e.*, $z_i \neq z_j$ for all $j < i$, the probability for o_i to be generated from a new cluster is then $\frac{\alpha}{\alpha+i-1}$.

The above generative process for cluster labels is also called Chinese Restaurant Process [2], since it simulates the table occupation by customers in Chinese restaurants. When the first customer comes to a restaurant, he will sit at an empty table; while the next customer will sit at a table if he knows someone there, with the probability proportional to the existing people in that table (the more people there, the higher probability he knows someone there); also there is a probability that he knows nobody in the current customers and sit at a new table with a probability $\frac{\alpha}{\alpha+i-1}$. Here, a table stands for a cluster. Therefore, the number of clusters are well modeled through this sampling process.

The larger α will result in more clusters, since the probability of creating a new cluster is larger then. In our model setting, we will set all α_t at different timestamps equally to α. Experiments in Sec. 5 show that α is not very sensitive for the cluster number.

4.1.2 Modeling Historical Impacts

Another constraint on the net-clusters is that the net-clusters at two consecutive timestamps should be similar to each other, namely, the evolution of net-clusters should be smooth. Meantime, we hope the same community could change insignificantly along the time. For example, authors' rank could be changing in the database community all the time. Therefore, we do not force different timestamp models to share the same clusters, as in [14, 25]. Instead, we design more flexible priors for each G_t, by considering its base measure as a Dirichlet distribution over mixture priors. In other words, this makes the empty tables not equally important, but have different hints denoted by different prior distributions. Namely, each net-cluster will have a prior label, either form a previous cluster $\theta^*_{k,t-1}$, or from background H_t. When a new object o_i comes, the generative process for him to join a cluster can be described as:

1. he chooses an existing cluster k with probability $\frac{n_k}{\alpha+i-1}$, where n_k is the number of objects in cluster k; or,
2. he chooses an empty cluster with probability $\frac{\alpha}{\alpha+i-1}$, then he chooses the prior knowledge for the cluster either as a previous cluster k_{t-1} with probability $(1-\lambda_t)\pi_{k,t-1}$, or as a background knowledge with probability λ_t, where $\pi_{k,t-1}$ is the learned mixture portions in G_{t-1} for cluster k_{t-1}, and λ_t can be learned empirically through the data.

Once the prior for each new net-cluster is chosen, if it is from a background model, the base measure $G_{0,t}(\mathbf{\Theta}_t)$ is drawn from Eq. 7; if it is from a previous net-cluster k_{t-1}, with parameter $\theta_{k,t-1}$, $G_{0,t}(\mathbf{\Theta}_t)$ is then defined as:

$$\begin{aligned}\theta^A &\sim Dirichlet(\beta_A\theta^A_{k,t-1}(a_1), \ldots, \beta_A\theta^A_{k,t-1}(a_{|A|_t}))\\ \theta^C &\sim Dirichlet(\beta_C\theta^C_{k,t-1}(c_1), \ldots, \beta_C\theta^C_{k,t-1}(c_{|C|_t}))\\ \theta^D &\sim Dirichlet(\beta_D\theta^D_{k,t-1}(d_1), \ldots, \beta_D\theta^D_{k,t-1}(d_{|D|_t}))\end{aligned} \tag{8}$$

where β_A, β_C and β_D is the precise of the Dirichlet distribution, which can be viewed as the pseudo objects already existing in a net-cluster with the defined prior knowledge. Notice that, for a set of target objects with size β, the sizes of objects from other types can be largely defined using the average degree of the target objects for each type. For example, in DBLP, a paper in average goes to 1 conference, has 2 authors, and contains 6 terms. In the experiment setting, we only need to set β_C, and set $\beta_A = 2\beta_C$ and $\beta_T = 6\beta_C$. Since β_C denotes the strength of the background model, the larger of it, the more smoothing between different clusters , and thus the fewer clusters.

4.2 Model Inference

For the first timestamp $t = 1$, our goal is to get $\mathbf{\Theta}^*_1$ that maximizes the posterior $p(\mathbf{\Theta}_1|\mathbf{O}_1)$, given H, α, β_C. For later timestamps $t > 1$, the goal is to derive $\mathbf{\Theta}^*_t$ that maximizes posterior $p(\mathbf{\Theta}_t|\mathbf{O}_t, \mathbf{\Theta}^*_{t-1})$, where $\mathbf{\Theta}^*_{t-1}$ is the best parameter derived in timestamp $t-1$. In this paper, we use a collapse Gibbs sampling following [11] to first sample the hidden variables $z_{i,t}$ for each object $o_{i,t}$, and then derive $\mathbf{\Theta}^*_t = \arg\max_{\mathbf{\Theta}_t} p(\mathbf{\Theta}_t|\mathbf{O}_t, \mathbf{Z}_t)$. Now we will introduce the inference algorithm step by step.

Step 1: Initialization.

For the input network G_t, if $t = 1$, then initialize the net-cluster set by partitioning the target objects into a temporary cluster number K_0 randomly. If $t > 1$, we initialize the net-cluster set by assigning them into $K_{t-1} + 1$ empty clusters with priors either from net-clusters in timestamp $t - 1$ or the background model H_t, according to model probability $p(o_{i,t}|k)$, where K_{t-1} is the cluster number in timestamp $t-1$. Also, assign λ_t as the proportion of objects with prior H_t over the whole target objects.

Step 2: Hidden Cluster Label Assignment.

Repeatedly sample each of the objects into existing clusters or new clusters, following the posterior distribution

$$p(z_{i,t}|o_{i,t}, \mathbf{z}_{-i,t}).$$

For simplicity, we now omit the sub-index t in the following unless they are necessary.

For each target object $o_i (i = 1, 2, \ldots, n)$, always considering it as the last observation,[4] the conditional distribution for its hidden cluster label z_i given all the other cluster labels, denoted as $\mathbf{z}_{-i}$, and the value of o_i integrating out $\mathbf{\Theta}$ can be derived as:

$$p(z_i = k|\mathbf{z}_{-i}, o_i) \propto p(z_i = k|\mathbf{z}_{-i}) \int_{\theta_k} p(o_i|z_i = k, \theta_k) \tag{9}$$

Specifically, there are four situations when sampling z_i:

(1.1) If k is a new cluster with background prior H, *i.e.*, $z_i \neq z_j$ for all $j \neq i$:

$$p(z_i \neq z_j(\forall j \neq i)|\mathbf{z}_{-i}, o_i, G_{0,H}) \propto \frac{\alpha\lambda_t}{n-1+\alpha} \int p(o_i|\theta)G_{0,H}(\theta)d\theta \tag{10}$$

where

$$\int p(o_i|\theta)G_{0,H}(\theta)d\theta = \int p(\mathbf{a}_i|\theta^A)p_{0,H}(\theta^A)d\theta^A \int p(\mathbf{c}_i|\theta^C)p_{0,H}(\theta^C)d\theta^C \int p(\mathbf{d}_i|\theta^D)p_{0,H}(\theta^D)d\theta^D$$

and according to the Dirichlet-multinomial conjugate, the posterior probability for $\mathbf{a}_i$ can be calculated as:

$$\int p(\mathbf{a}_i|\theta^A)p_{0,H}(\theta^A)d\theta^A = \frac{\Gamma(\beta_A)}{\Gamma(\beta_A + n_{a_i})} \frac{\prod_{j=1}^{|A|}\Gamma(\frac{\beta_A}{|A|} + n_{a_{ij}})}{\prod_{j=1}^{|A|}\Gamma(\frac{\beta_A}{|A|})} \tag{11}$$

where n_{a_i} is the total number of authors for paper o_i and $n_{a_{ij}}$ is the number of author a_j in paper o_i, and which are similar for $\mathbf{c}_i$ and $\mathbf{d}_i$.

(1.2) If k is a new cluster with net-cluster prior $\theta_{k,t-1}$:

$$\begin{aligned} &p(z_i \neq z_j(\forall j \neq i)|\mathbf{z}_{-i}, o_i, G_{0,k_{t-1}}) \\ \propto &\frac{\alpha(1-\lambda_t)\pi_{k,t-1}}{n-1+\alpha} \int p(o_i|\theta)G_{0,k_{t-1}}(\theta)d\theta \end{aligned} \tag{12}$$

with the posterior probability for $\mathbf{a}_i$ be calculated as:

$$\int p(\mathbf{a}_i|\theta^A)p_{0,k_{t-1}}(\theta^A)d\theta^A = \frac{\Gamma(\beta_A)}{\Gamma(\beta_A + n_{a_i})} \frac{\prod_{j=1}^{|A|}\Gamma(\beta_A\theta^A_{k,t-1}(j) + n_{a_{ij}})}{\prod_{j=1}^{|A|}\Gamma(\beta_A\theta^A_{k,t-1}(j))} \tag{13}$$

(2.1) If k is an existing cluster with background prior H, *i.e.*, $z_i = z_j$ for some $j \neq i$:

$$p(z_i = k|\mathbf{z}_{-i}, o_i, G_{0,H}) \propto \frac{n_k^{-i}}{n-1+\alpha} \int p(o_i|\theta_k)p(\theta_k|\mathbf{z}_{-i}, \mathbf{O}_{-i}, G_{0,H})d\theta_k \tag{14}$$

where n_k^{-i} stands for the number of times that label $z = k$ has been assigned, excluding the current (i.e., the i-th) instance; $p(\theta_k|\mathbf{z}_{-i}, \mathbf{O}_{-i}, G_{0,H})$ is a posterior distribution given current observation in cluster k without o_i, and the prior distribution $G_{0,H}$. The component probability $p(\mathbf{a}_i)$ in the formula should be

$$\begin{aligned} &\int p(\mathbf{a}_i|\theta^A)p_{0,H}(\theta^A)d\theta^A \\ =&\frac{\Gamma(\beta_A + n_{k,A}^{-i})}{\Gamma(\beta_A + n_{a_i} + n_{k,A}^{-i})} \frac{\prod_{j=1}^{|A|}\Gamma(\frac{\beta_A}{|A|} + n_{a_{ij}} + n_{k,A}^{-i}(j))}{\prod_{j=1}^{|A|}\Gamma(\frac{\beta_A}{|A|} + n_{k,A}^{-i}(j))} \end{aligned} \tag{15}$$

where $n_{k,A}^{-i}$ is the total author number in cluster k without paper o_i and $n_{k,A}^{-i}(j)$ is the number of author a_j in cluster k without o_i, and which are similar for other types of objects.

(2.2) If k is an existing cluster with net-cluster prior $\theta_{k,t-1}$, *i.e.*, $z_i = z_j$ for some $j \neq i$:

$$\begin{aligned} &p(z_i = k|\mathbf{z}_{-i}, o_i, G_{0,k_{t-1}}) \\ \propto &\frac{n_k^{-i}}{n-1+\alpha} \int p(o_i|\theta_k)p(\theta_k|\mathbf{z}_{-i}, \mathbf{O}_{-i}, G_{0,k_{t-1}})d\theta_k \end{aligned} \tag{16}$$

The component probability $p(\mathbf{a}_i)$ in the formula should be

$$\begin{aligned} &\int p(\mathbf{a}_i|\theta^A)p_{0,k_{t-1}}(\theta^A)d\theta^A \\ =&\frac{\Gamma(\beta_A + n_{k,A}^{-i})}{\Gamma(\beta_A + n_{a_i} + n_{k,A}^{-i})} \frac{\prod_{j=1}^{|A|}\Gamma(\beta_A\theta^A_{k,t-1}(j) + n_{a_{ij}} + n_{k,A}^{-i}(j))}{\prod_{j=1}^{|A|}\Gamma(\beta_A\theta^A_{k,t-1}(j) + n_{k,A}^{-i}(j))} \end{aligned} \tag{17}$$

Notice that, if the only object for some cluster is assigned to other clusters, this cluster should be then removed. The Gibbs sampling process described are repeated multiple times and the average value for the assignment is used to estimate the expectation of the real distribution of hidden variables. Since different cluster numbers may be obtained in different samples (only occasionally, when the sampling converges to real distribution), we use hierarchical clustering to merge clusters into the average cluster number.

Step 3: Cluster Parameter Estimation.

Once the assignment for each object is fixed, the parameter θ_k for each cluster can be estimated accordingly. Specifically, each component of $\theta_k = (\theta_k^A, \theta_k^C, \theta_k^D)$ is a Dirichlet distribution given the observations in cluster k, and has the MLE estimation as

$$\theta_k^A(j) = \frac{\beta_A/|A| + n_j^A}{\beta_A + n^A}; \theta_k^C(j) = \frac{\beta_C/|C| + n_j^C}{\beta_C + n^C}; \theta_k^D(j) = \frac{\beta_D/|D| + n_j^D}{\beta_D + n^D} \tag{18}$$

if the prior for the net-cluster is from background model H; otherwise, the MLE estimation is

$$\begin{aligned} \theta_k^A(j) &= \frac{\beta_A\theta_{k,t-1}(j) + n_k^A(j)}{\beta_A + n_k^A}; \\ \theta_k^C(j) &= \frac{\beta_C\theta_{k,t-1} + n_k^C(j)}{\beta_C + n_k^C}; \\ \theta_k^D(j) &= \frac{\beta_D\theta_{k,t-1} + n_k^D(j)}{\beta_D + n_k^D} \end{aligned} \tag{19}$$

if the prior is from a previous net-cluster $\theta_{k,t-1}$. They are exactly the simple ranking used in [18] and [19] smoothed with background model. Other empirical parameter estimation methods can also be used here, for example, the

[4] Due to the exchangeability of the data, this assumption does not affect the model result.

```
**Input**: Network $G_t$, $\mathbf{\Theta}^{(t-1)}$; $\alpha$, $\beta_C$ ;
**Output**: The net-clusters $\{G_{k,t}\}$; the parameters
        $\mathbf{\Theta}_t = \{\theta_t^A, \theta_t^C, \theta_t^D\}$;
Initialize the net-clusters either using random partitioning
for the first time, or according to the conditional probability
for each prior;
**repeat**
    %Assign the hidden cluster labels;
     1. Calculate the posterior probability for each $o_i$ in
    each existing cluster $k$ and new cluster $k+1$ ;
     2. Sampling the object to the $k$, if it is a new cluster,
    $k++$, and record its prior distribution ;
     3. if net-cluster $k_{old}$ for $o_i$ contains no objects, remove
    the cluster;
**until** *reaches iteration number*;
Extract net-clusters $\{G_{k,t}\}$ and estimate their parameters
$\mathbf{\Theta}_t$;
```

Algorithm 1: Model inference algorithm.

authority ranking used in [18] and [19]. The posterior probability that each object is belonging to each cluster can also be calculated.

The inference algorithm is summarized as in Algorithm 1. In our experiment setting, the Gibbs sampling iteration number is set as 1500, the burning period is 1000, and the sample gap is 10.

5. EXPERIMENT

In this section, we will study the effectiveness of the evolutionary net-cluster model using real datasets.

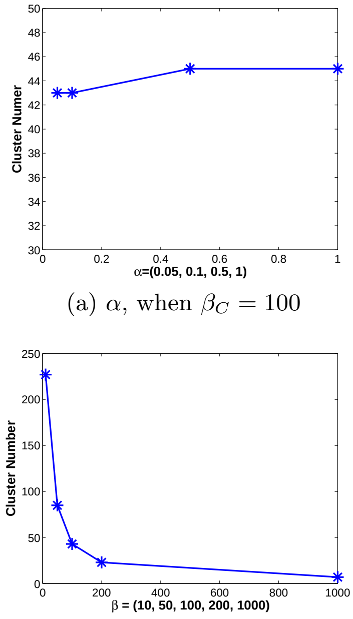
5.1 Dataset

We first use the DBLP data to extract the network sequences between the years 1990 and 2008, where authors publishing more than 10 papers and conferences containing more than 200 papers in this period are kept. The network contains four types of objects, namely papers, authors, conferences(journals) and terms, and follows the star network schema. Terms are extracted from paper titles. Stop words and low frequency terms (less than 20) are removed. The papers associated with each timestamp range from 8K to 65K.

The second dataset is extracted from Delicious.com, from Jan. 1 to Jan. 28, 2010. The network also contains four types of objects, namely tagging events, users, websites and tags, following star network schema. Users, websites and tags appearing less than twice are removed. The tagging events are grouped into four weeks according to their tagging time, and thus a sequence of four consecutive networks are obtained.

5.2 Parameter Setting Study

There are two parameters defined in Evo-Netclus model, namely α and β_C, where α defines the cluster structure probability, and β_C controls the smoothness between different net-clusters. We use the DBLP network of year 1991 as the test network, which contains 9574 papers, to study the relation between cluster number and parameters α and β_C (See Fig. 3). As expected, larger α will result in more cluster numbers. However, the impact is not that sensitive. Also, lager β_C leads to smaller cluster numbers, as the strength of smoothing between different net-clusters grows. Actually, difference β_C can provide different scale's view of cluster evolution. In our experiments, we set $\alpha = 0.1$ and $\beta_C = 1000$.



(a) α, when $\beta_C = 100$

(b) β_C, when $\alpha = 0.1$

Figure 3: Parameter Study: α and β_C vs. Cluster Number

5.3 Comparative Performance Study

As we claimed, there are two advantages of Evo-NetClus model, first is that it can consider different types of objects, and second is that each clustering results can carry historical information. We now compare Evo-NetClus with three other degenerated clustering models using DBLP data, namely using fewer types of objects or not using historical priors, by evaluating the similarity compactness of papers within the same conference and among different conferences on the test dataset. We define similarity compactness of papers considering conferences as $ConfCompact$:

$$ConfCompact = \frac{\sum_k \sum_{o_i \in k} s(o_i, c_k)/|O|}{\sum_k \sum_{l \neq k} s(c_k, c_l)/(k(k-1))}$$

where o_i is considered as a $(n+1)_{th}$ object and can be represented as a vector of posterior probability given each existing net-cluster in the training model (Eq. 13-16), c_k is the center of conference k, and $s(.,.)$ denotes some similarity function, where we use cosine similarity. Intuitively, the papers in the same conference should be more similar than in different conferences. Therefore, the larger the value of $ConfCompact$, the better the posterior probability vector is, and the better the model is. To overcome the sparsity of conferences in the test data, we only use top-5 conferences containing most testing papers into within-conference similarity calculation. From Table 1, we can see that Evo-NetClus that considers both multiple types of objects and historical impact information indeed gives the best performance for both test datasets, in terms the ability to provide better similarity feature given by the training model. By

Year	Training Type	Testing Type	Test Size 10% (cluster number K)	Test Size 20% (cluster number K)
1992	Term	Term	1.600 (4)	1.390 (4)
1992	Term+Author	Term+Author	2.205 (8)	1.697 (6)
1992	Term+Author+Conf.	Term+Author	2.434 (8)	2.095 (8)
1992|1991	Term+Author+Conf.	Term+Author	**2.8365** (8)	**2.671** (8)

Table 1: Conference Compactness of Different Models on Test Dataset

the comparison between Row 2 and Row 3, we can see that additional type, *i.e.*, conference, in training dataset can enhance the ranking for author and term type, and derive better performance on test dataset where only author and term type are used. From the comparison between the last two rows, we can see that Evo-NetClus model considering historical information gives much better results, which means independently deriving net-clusters at each timestamp, and then construct evolution structure between different timestamps may not be a good option, since the net-clusters thus derived will be overfitting the current dataset. The cluster numbers generated by each algorithm for each training data set are shown in Table 1 as well.

Perplexity is a measure for evaluating the goodness for statistical models, which is defined as:

$$perplexity = 2^{-\sum_{i=1}^{N}\frac{1}{N}\log_2 q(o_i)}$$

where $q(o_i)$ is the estimated probability for object o_i determined by the mixture model defined in Eq. 1 and Eq. 6, using estimated parameters obtained by Eq. 18 and 19. From perplexity comparison between the two models in Table 2, we can similarly find that Evo-NetClus with historical impact (Row 2) has lower perplexity for both testing datasets, which means including historical impacts indeed enhances the performance of the model.

5.4 Case Study

We now show a small portion (see Fig. 4) of net-clusters derived from DBLP dataset, and their evolution structure via priors, through year 1991-2000 (only the year 1991, 1994, 1997 and 2000 are shown), which is generated using parameter $\alpha = 1$ and $\beta_C = 1000$. For each net-cluster, top-5 conferences and top-5 terms are output. A link exists between two net-clusters, if the previous one is the prior of the latter. Notice that, using the prior dependency, we can read the split, birth and death easily. For cluster merge, it still needs some postprocessing effort to judge whether a cluster disappears because of a real death or because of a merge into other clusters. From the case study, we can see that the data mining community is evolved from the database community in the year 2000. Notice that, though KDD conference first appeared in the year 1995, but the community had not been well formed until other data mining conferences appeared, such as PKDD, PAKDD and so on.

Another case study is from Delicious (see Fig. 5), which is generated using parameter $\alpha = 1$ and $\beta_{url} = 500$. Three communities are shown, which corresponding to politics, apple, and world news. Fig. 5(a) shows the top-10 tags used in each community, and Fig. 5(b) shows the trends of the size of the related tagging events for each community. For example, Apple announced its new product "iPad" on Jan. 27, which results in the significantly increasing of the related tagging events happened in Apple community in the fourth week of January.

6. CONCLUSIONS

In this paper, we study the problem of detecting evolutionary net-clusters, *i.e.*, clusters defined on heterogeneous networks, which are able to describe a community that contains different types of objects. A Dirichlet Process Mixture Model-based generative model Evo-NetClus is proposed to model the net-cluster generation process with the time information. At each time stamp, a net-clustering with the best cluster number and smoothing with net-clusters at previous time stamps are automatically detected. A Gibbs sampling-based algorithm is proposed to inference model. Experiments on DBLP and Delicious dataset have shown the effectiveness of the algorithm.

7. REFERENCES

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, 2008.

[2] D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. Aug 2009.

[3] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML '06*, pages 113–120, New York, NY, USA, 2006. ACM.

[4] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *KDD '07*, pages 153–162, New York, NY, USA, 2007. ACM.

[5] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD '01*, pages 269–274, New York, NY, USA, 2001. ACM.

[6] W. Fu, L. Song, and E. P. Xing. Dynamic mixed membership blockmodel for evolving networks. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 329–336, New York, NY, USA, 2009. ACM.

[7] M. S. Handcock, A. E. Raftery, and J. M. Tantrum. Model-based clustering for social networks. *Journal Of The Royal Statistical Society Series A*, 170(2):301–354, 2007.

[8] P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION*, 97:1090–1098, 2001.

[9] M.-S. Kim and J. Han. A Particle-and-Density Based Evolutionary Clustering Method for Dynamic Networks. In *2009 Int. Conf. on Very Large Data Bases*, , Lyon, France, August 2009.

[10] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *KDD '05*, pages 198–207, New York, NY, USA, 2005. ACM.

Year	Training Type	Testing Type	Test Size 10%	Test Size 20%
1992	Term+Author+Conf.	Term+Author+Conf.	3.493×10^{18}	4.673×10^{18}
1992|1991	Term+Author+Conf.	Term+Author+Conf.	$\mathbf{6.384 \times 10^{17}}$	$\mathbf{7.106 \times 10^{17}}$

Table 2: Perplexity Comparison between Models with/without Historical Prior

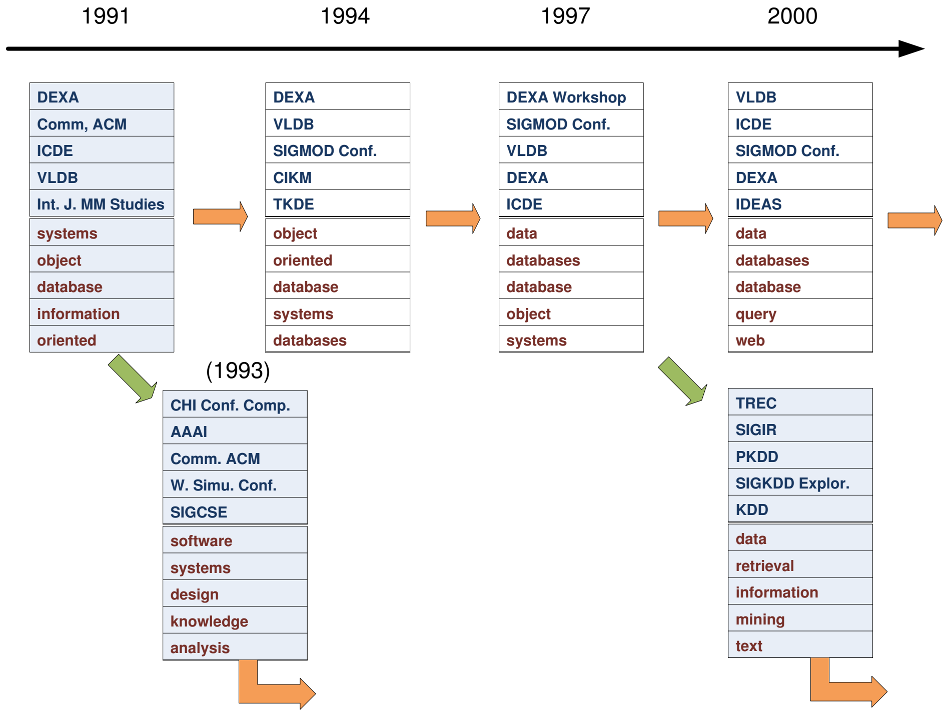


Figure 4: Case Study 1: Evolutionary Communities in DBLP

[11] R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, June 2000.

[12] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74:036104, 2006.

[13] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2), 2004.

[14] L. Ren, D. B. Dunson, and L. Carin. The dynamic hierarchical dirichlet process. In *ICML '08*, pages 824–831, New York, NY, USA, 2008. ACM.

[15] P. Sarkar and A. W. Moore. Dynamic social network analysis using latent space models. *SIGKDD Explor. Newsl.*, 7(2):31–40, 2005.

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. In *CVPR'97*, page 731, Washington, DC, USA, 1997. IEEE Computer Society.

[17] T. A. B. Snijders. Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 3, 2002.

[18] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. Rankclus: integrating clustering with ranking for heterogeneous information network analysis. In *EDBT '09*, pages 565–576, New York, NY, USA, 2009. ACM.

[19] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD '09*, pages 797–806, New York, NY, USA, 2009. ACM.

[20] L. Tang, H. Liu, J. Zhang, and Z. Nazeri. Community evolution in dynamic multi-mode networks. In *KDD '08*, pages 677–685, New York, NY, USA, 2008. ACM.

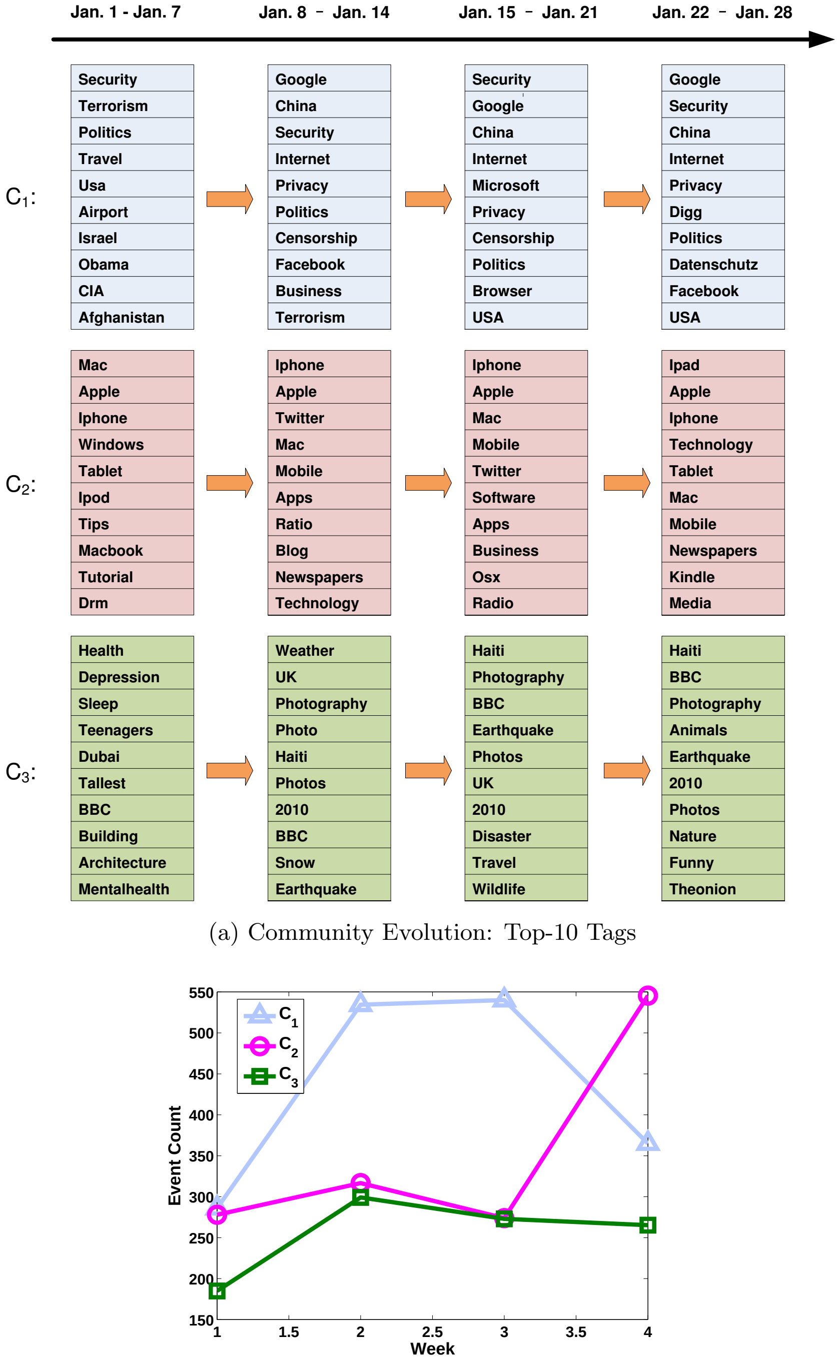
[21] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[22] U. von Luxburg. A tutorial on spectral clustering. Technical report, Max Planck Institute for Biological Cybernetics, 2006.

[23] S. White and P. Smyth. A spectral clustering approach to finding communities in graph. In *SDM '05*, 2005.

[24] T. Xu, Z. M. Zhang, P. S. Yu, and B. Long. Dirichlet process based evolutionary clustering. In *ICDM '08*, pages 648–657, Washington, DC, USA, 2008. IEEE Computer Society.

[25] T. Xu, Z. M. Zhang, P. S. Yu, and B. Long. Evolutionary clustering by hierarchical dirichlet process with hidden markov state. In *ICDM '08*, pages 658–667, Washington, DC, USA, 2008. IEEE Computer Society.



	Jan. 1 - Jan. 7	Jan. 8 - Jan. 14	Jan. 15 - Jan. 21	Jan. 22 - Jan. 28
C_1:	Security	Google	Security	Google
	Terrorism	China	Google	Security
	Politics	Security	China	China
	Travel	Internet	Internet	Internet
	Usa	Privacy	Microsoft	Privacy
	Airport	Politics	Privacy	Digg
	Israel	Censorship	Censorship	Politics
	Obama	Facebook	Politics	Datenschutz
	CIA	Business	Browser	Facebook
	Afghanistan	Terrorism	USA	USA
C_2:	Mac	Iphone	Iphone	Ipad
	Apple	Apple	Apple	Apple
	Iphone	Twitter	Mac	Iphone
	Windows	Mac	Mobile	Technology
	Tablet	Mobile	Twitter	Tablet
	Ipod	Apps	Software	Mac
	Tips	Ratio	Apps	Mobile
	Macbook	Blog	Business	Newspapers
	Tutorial	Newspapers	Osx	Kindle
	Drm	Technology	Radio	Media
C_3:	Health	Weather	Haiti	Haiti
	Depression	UK	Photography	BBC
	Sleep	Photography	BBC	Photography
	Teenagers	Photo	Earthquake	Animals
	Dubai	Haiti	Photos	Earthquake
	Tallest	Photos	UK	2010
	BBC	2010	2010	Photos
	Building	BBC	Disaster	Nature
	Architecture	Snow	Travel	Funny
	Mentalhealth	Earthquake	Wildlife	Theonion

(a) Community Evolution: Top-10 Tags

(b) Community Size Evolution

Figure 5: Case Study 2: Evolutionary Communities in Delicious

[26] H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Bipartite graph partitioning and data clustering. In *CIKM '01*, pages 25–32, New York, NY, USA, 2001. ACM.

[27] X. Z. Zoubin, X. Zhu, Z. Ghahramani, and J. Lafferty. Time-sensitive dirichlet process mixture models. Technical report, 2005.